|  |  |  |
| --- | --- | --- |
| **Use Case 1: Flow of Events for the *Initialise Game* use-case** | | |
| **Objectives** | **To allow players to set up and begin a game.** |
| **Actor(s)** | Player. |
| **Precondition** | The programme has been launched. |
| **Main Flow** | 1. The menu is displayed in the console when the game is launched. 2. The player enters option ‘2’ to start new game. 3. A dialogue is displayed. 4. The player is prompted to enter the number of players. 5. The player enters a whole number 2-4. 6. The player is prompted to enter a player’s name. 7. The player enters a name. 8. Steps 5 and 6 are repeated for the number of players enter in step 4. 9. A dialogue is displayed. |
| **Alternative Flow** | **At 2.**  The player enters option ‘1’ to display rules.  The information is displayed.  The player is prompted ‘To return to the main menu press [Enter]’.  **At 2.**  The player enters option ‘3’ to restore a saved game.  The game resumes from the last save point.  **At 2.**  The player enters option ‘4’ to enable speech.  The player is asked to enter Y or N to enable speech.  The player enters ‘Y’ and speech is enabled.  The menu is displayed again and spoken, with option 4 being replaced with the option to disable speech.  The player enters ‘N’ and is returned to starting menu.  **At 4.**  If anything, other than ‘2, 3, or 4’ is typed into the console an appropriate error message is displayed and the question is repeated until an accepted character is entered.  **At 6.**  The player enters a name outside the accepted parameters.  An appropriate error message is displayed, and the player is prompted to enter their name again.  **At 6.**  The player enters name already entered by a previous player.  An appropriate error message is displayed, and the player is prompted to enter their name again. |
| **Post-condition** | The game is started. |

|  |  |  |
| --- | --- | --- |
| **Use Case 2: Flow of Events for the *Move* use-case** | | |
| **Objectives** | **To move the player from their current position on the game board by the number randomly generated.** |
| **Actor(s)** | Player |
| **Precondition** | It is the players turn. |
| **Main Flow** | 1. Virtual dice are rolled, and a number randomly generated. 2. The player is told the total value. (For example, “[Player name] has rolled 10.” 3. Using the randomly generated value the current player is moved a specified number of squares on the virtual game board. 4. The players position on the board is temporarily saved. 5. A dialogue is displayed telling the current player what square they landed on and all relevant information about that square. 6. Depending on what square the player passes or lands on the Pass Go, Purchase Element, Pay Rent uses cases may be performed. 7. A bankrupt check is completed to ensure the current player has at least 0 resources. 8. A menu is displayed. 9. The current player enters option ‘5’ to end their turn. |
| **Alternative Flow** | **At 9.**  The current player enters option ‘1’ to display all systems.  All systems are displayed with the system name and the elements they contain.  **At 9.**  The current player enters option ‘2’ to Display Resources & Properties Owned.  The resources and elements the current player owns are displayed.  **At 9.**  The current player enters option ‘3’ to Develop an element.  Develop Element use-case is performed.  **At 9.**  The current player enters option ‘4’ to Trade an element.  Trade Element use-case is performed.  **At 9.**  The current player enters option ‘6’ to Save Game.  The game is saved.  **At 9.**  The current player enters option ‘7’ to Quit the game.  Game Over use-case is performed.  **At 9.**  The current player enters option ‘8’ to display the board layout.  The board layout is displayed. |
| **Post-condition** | Player’s move ends and Move is repeated for the next Player. |

|  |  |  |
| --- | --- | --- |
| **Use Case 3: Flow of Events for the *Pass Go* use-case** | | |
| **Objectives** | **The current player has passed ‘GO’ and receives additional resources.** |
| **Actor(s)** | Player |
| **Precondition** | It’s the players turn.  The player has rolled the dice.  The player has passed or landed on the ‘GO’ element. |
| **Main Flow** | 1. A dialogue is displayed telling the current player they have passed go and received resources. 2. The appropriate resources are added to the current player. |
| **Alternative Flow** | None. |
| **Post-condition** | The current players resources are updated. |

|  |  |  |
| --- | --- | --- |
| **Use Case 4: Flow of Events for the *Purchase Element* use-case** | | |
| **Objectives** | **To purchase an Element.** |
| **Actor(s)** | Player |
| **Precondition** | It’s the players turn.  The player lands on an element not owned by another player. |
| **Main Flow** | 1. The current player is asked “[Player name] Do you want to purchase ‘Element X’? Enter Y / N”. 2. The player types Y into the console. 3. The appropriate resources are removed from the player. 4. The players’ name is assigned to ‘Element X’. |
| **Alternative Flow** | **At 2.**  The current player types ‘N’ into the console, choosing not to purchase ‘Element X’.  The square is not purchased by the current player and their resources are not changed.  The current player is asked if they want to offer the square to another player.  The current player enters ‘Y’ and is prompted to select a player by a corresponding number.  A number is entered, and the selected player is asked “[Player name] Do you want to purchase ‘Element X’? Enter Y / N”.  The selected player enters ‘Y’.  The appropriate resources are removed, and the players name assigned to ‘Element X’.  The selected player enters ‘N’.  This alternative flow is repeated until a player either accepts the purchase or cancels.  The current player enters ‘N’.  An appropriate message is displayed.  **At 2**.  If anything, other than ‘Y’ or ‘N’ (not case sensitive) is typed into the console an appropriate error message is displayed and the question is repeated until an accepted character is entered.  **At 3.**  The player does not have sufficient resources to purchase ‘Element X’. An appropriate message is displayed, no one is assigned to ‘Element X’ and no resources change. |
| **Post-condition** | The players resources are updated and ‘Elements’ are updated. |

|  |  |  |
| --- | --- | --- |
| **Use Case 5: Flow of Events for the *Develop Element* use-case** | | |
| **Objectives** | **To develop an Element within a System the current player owns.** |
| **Actor(s)** | Player |
| **Precondition** | It’s the players turn.  The player has moved.  The player owns all elements within a System.  The element is not fully developed. |
| **Main Flow** | 1. The current player enters option ‘3’ to develop an element 2. A development menu is displayed with all elements available for the current player to develop with a corresponding number. 3. The current player enters the number for the element to develop. 4. The appropriate resources are removed from the current player. 5. The development level of ‘Element X’ is increased along with the cost of rent. 6. A dialogue detailing the development taking place is displayed. 7. The development menu is displayed until all elements available for development are fully developed or the current player chooses to cancel. |
| **Alternative Flow** | **At 2.**  The current player does not own a complete system of elements and an appropriate message is displayed.  **At 3**.  If invalid input is entered an appropriate message is displayed and the question is repeated until valid input is entered.  **At 4.**  The player does not have sufficient resources to develop ‘Element X’. An appropriate message is displayed, ‘Element X’ is not developed, and no resources change. |
| **Post-condition** | The current players resources and ‘Elements’ are updated. |

|  |  |  |
| --- | --- | --- |
| **Use Case 6: Flow of Events for the *Pay Rent* use-case** | | |
| **Objectives** | **To pay rent on an Element owned by another player.** |
| **Actor(s)** | Player |
| **Precondition** | It’s the players turn.  The player lands on an Element owned by another player. |
| **Main Flow** | 1. The player who owns ‘Element X’ is asked “[Player name] Do you want to charge [Player name] rent for ‘Element X’? Enter Y / N”. 2. The player types ‘Y’ into the console. 3. The appropriate resources are removed from the current player. 4. The equivalent resources are added to the player who owns ‘Element X’. |
| **Alternative Flow** | **At 2.**  The player who owns ‘Element X’ types ‘N’ into the console, choosing not to charge rent.  The current player resources are not changed.  **At 2**.  If invalid input is entered an appropriate message is displayed and the question is repeated until valid input is entered.  **At 3.**  The current player does not have sufficient resources to pay rent for ‘Element X’. The current player has negative resources and Game Over use case is performed. |
| **Post-condition** | The current players resources are updated. The player who owns ‘Element X’ resources are updated. |

|  |  |  |
| --- | --- | --- |
| **Use Case 7: Flow of Events for the *Trade Element* use-case** | | |
| **Objectives** | **To trade an Element the current player owns to another player.** |
| **Actor(s)** | Player |
| **Precondition** | It’s the players turn.  The player has moved.  The player owns the element. |
| **Main Flow** | 1. The current player enters option ‘4’ to trade an element 2. A trade menu is displayed with all elements available for the current player to trade with a corresponding number. 3. The current player enters the number for the element to trade. 4. A menu showing the other players with a corresponding number is shown for the current player to choose who to trade with. 5. The current player enters the number for the player to trade with. 6. The selected player is asked “[Player name] would you like to buy ‘Element X’ from [Player name] Enter Y / N”. 7. The selected player enters ‘Y’. 8. The appropriate resources are removed from the selected player and added to the current player. 9. The selected players name replaces the current players name as owner of ‘Element X’. 10. A dialogue detailing the trade taking place is displayed. 11. The trade menu is displayed until there are no elements available for trading or the current player chooses to cancel. |
| **Alternative Flow** | **At 2.**  The current player does not own any elements and an appropriate message is displayed.  **At 3**.  If invalid input is entered an appropriate message is displayed and the question is repeated until valid input is entered.  **At 5**.  If invalid input is entered an appropriate message is displayed and the question is repeated until valid input is entered.  **At 7.**  The selected player enters ‘N’, choosing not to purchase ‘Element X’.  The current players’ resources and elements are not changed.  The selected players’ resources and elements are not changed.  **At 8.**  The selected player does not have sufficient resources to purchase ‘Element X’. An appropriate message is displayed. |
| **Post-condition** | The current players resources and ‘Elements’ are updated.  The selected players resources and ‘Elements’ are updated. |

|  |  |  |
| --- | --- | --- |
| **Use Case 8: Flow of Events for the *Random Event* use-case** | | |
| **Objectives** | **To generate a random event (positive or negative), which adds or removes resources appropriately for a randomly chosen player** |
| **Actor(s)** | Player |
| **Precondition** | All players have completed a turn. |
| **Main Flow** | 1. A dialogue is displayed to indicate an event will be generated 2. A dialogue is displayed detailing a positive event 3. A dialogue is displayed congratulating a randomly selected player 4. Appropriate randomly generated resources are added to the selected player’s balance 5. A dialogue is displayed indicating the end of the event |
| **Alternative Flow** | **At 2.**  A dialogue is displayed detailing a negative event.  A dialogue is displayed indicating a randomly selected player will be fined  Appropriate randomly generated resources are removed from the selected player’s balance  Main flow continues at step 5. |
| **Post-condition** | The selected player’s resources are updated |

|  |  |  |
| --- | --- | --- |
| **Use Case 9: Flow of Events for the *Game Over* use-case** | | |
| **Objectives** | **To conclude the game.** |
| **Actor(s)** | Player |
| **Precondition** | None. |
| **Main Flow** | 1. All systems have been fully developed. 2. An epilogue is displayed detailing the launch. 3. A dialogue is displayed detailing the game state. |
| **Alternative Flow** | **At 1.**  The current player enters option ‘7’ to quit the game.  Main flow continues at step 3.  **At 1.**  The current player has a negative amount of resources.  Main flow continues at step 3. |
| **Post-condition** | The game is over. |